

Waveform Collection

15Hz Quicker Digitizer

Thu, Nov 11, 1999

In preparation for Run II, it will be necessary to study the problems of operating both Linac and Booster at 15Hz. This should include access to waveforms collected by the new Quicker digitizers that can operate at digitize rates of up to 20 MHz. This note describes a scheme for collecting such waveforms at rates of up to 15Hz, or as fast as the Quicker digitizer can operate.

The current speed of the Quicker digitizer is limited by the time required to store all 8 waveforms in its memory. Each waveform allows for 16K data points, so this means storing a total of 256K bytes of memory. The digitizers write into FIFOs first, one for each channel. The FIFOs are emptied into memory as fast as possible. The time required for storing all 8 waveforms is about 100 ms. The speed of writing to memory for each channel is about 16K points/100 ms, or about 160 points/ms. If one wished to collect 10MHz data for 30 μ s, this would mean 300 points. About 2 ms would be enough time to store 300 points for each of the 8 channels. Accesses to memory must be interleaved between channels. The software cannot retrieve the data until it has been stored into memory, so it is important to be concerned with this step.

Data stream approach

There is support in the system software for a data access table entry type that copies blocks of memory into a data stream. The fields in this entry specify the target data stream number, the base address of the memory to be recorded into the data stream, the number of bytes of memory to be recorded, and for the case of multiple blocks of memory, a delta memory size and a count. What is recorded into the data stream for each memory block is a 16-byte header followed by the block of data. The method for doing this is to copy the header and the data from the source block onto the stack, then call DSWrite to write the header+data into the data stream as a single record. Notice that this method requires copying the data block twice, in order to make use of the services provided by DSWrite.

To use the above scheme, the data stream must be defined knowing the maximum size of an entry. Suppose that it was planned to record 600 bytes of waveform data (300 points) for each of 4 waveforms. The data stream could be defined with a fixed record size of 616 bytes, which allows space for the 16-byte header plus the 600-bytes of data. Its total size would not have to be too large, but it has to be at least as large as necessary to hold the number of waveforms to be recorded, maybe plus one more than that.

The 16-byte header structure indicates the time-of-day in the usual 8-byte BCD format, the 4-byte memory address from which the data in the record was copied, and a 4-byte filler. The format of the time-of-day is as in the following example:

9911 0513 4322 1000,

which means November 5, 1999 at 1343:22–10, where the last value indicates the 15Hz cycle of the current second, in the range 00–14. All data stream records recorded on a given 15Hz cycle will have the same time-of-day value.

To access this data via the data acquisition protocol, the listype 51 may be used with 15Hz periodic replies. The number of bytes requested must be enough to "keep up" with the rate of recording data stream records, plus 4 bytes for a special header, which consists of a count word and a word that indicates the record size. So, a requester will receive a structure that includes the 4-byte header following by some number of data stream records, each of which includes a 16-byte header.

The digitizer hardware must be set up to operate by itself in order to collect such data as described above. The data access entry can be conditioned by a Bit# using a type 7F periodicity entry type. The Bit# used may reflect a clock event number, or it could be the result of a local application's logic performed before the waveform capture entry. One may wish to enable the storing of the 3rd event following a given reset event, for example. The logic to do this can be supported by a special local application written for the purpose.

As to what can be done with the data collected for a Vax console application, it may be plotted and/or stored into a file for later retrieval and analysis. The data stream access merely provides the waveform data with time stamps and memory addresses that can identify which signal is represented. Additional information can of course be stored by the application, such as the Acnet device name, etc.

Simple 15Hz access

An alternative approach to collecting 15Hz waveforms in a simple way is to use the RETDAT protocol, with an event-specified FTD. The waveforms must be operating in a continuous hardware mode, without using the SWFT local application that prevents it from achieving 15Hz operation. The listype used could specify memory word access, in which case the SSDN must include the hardware waveform memory address. Alternatively, a new listype could be designed that uses a Channel number ident format. The support software would have to look up the waveform memory address in the CINFO table, then return the data beginning at that address. It is important that the time of fulfilling the data request is late enough so that the number of points requested are already stored in the waveform memory. In the Linac nodes, this is easy, because the request fulfillment time is delayed by waiting for data from the SRMs. For an IRM, one needs to look at this more critically. For Booster, with the 15Hz cycle as seen by the IRM beginning at 35 ms into the accelerator cycle, any waveforms captured early in the cycle will long since have been stored. It would seem that in many cases, the required condition is fulfilled, so that every time an event occurs, the waveform will be digitized and delivered to the RETDAT requester. The Vax console application would need to do something with the data it received. It also would have to keep up with 15Hz reception of replies.